Some new results on recursive aggregation rules

Daniel Gómez
Escuela de Estadística
Universidad Complutense
Madrid, 28040 Spain
dagomez@estad.ucm.es

Javier Montero
Facultad de Matemáticas
Universidad Complutense
Madrid, 28040 Spain
javier_montero@mat.ucm.es

Abstract

As pointed by Cutello and Montero in a previous paper, consistency of an aggregation rule based upon a sequence of binary operators can be justified from an *operational* argument, by imposing a recursive calculus. Following this *recursive* approach, it was later proven that under certain regularity conditions, strict increasingness leads to quasi-additive solutions. In this paper we propose an alternative and more general result, avoiding some of those regularity conditions. Moreover, we point out that in practice we should evaluate only those aggregations being allowed by the decision maker. Preference structures are considered as an illustrative example.

Keywords: Aggregation functions, Recursive rules, Fuzzy Sets.

1 Introduction.

Aggregation models play a relevant role in decision making. In fact, most decision making problems require some aggregation technique in order to help decision makers to understand the information they are given. If we talk about remote sensing, for example, a typical objective it to classify pixels (units of the land surface) within homogeneous classes (see, e.g., Amo *et al.* [2, 3, 4, 5]). This particular image classification problem implies for each pixel a big amount of data, and data dimension needs to be reduced in order to be managed by the decision maker. In addition, since classification of each pixel should also take into account behavior in its respective neighborhood, information relative to surrounding pixels should also be aggregated. These aggregation processes are usually solved by means of a sequential procedure, but it is obvious that we can not restrict to a unique formula, to be applied again and again no matter the context. Moreover, we realize that in some cases data show a particular and informative structure (e.g., the neighborhood in a surface is not the neighborhood in the real space).

Aggregation procedures are defined as *rules* that tell us how to proceed with the information reaching to us, no matter if its dimension is previously known. From this fact, Cutello–Montero [9] have claim that *consistency* of such a rule can be guaranteed from an *operational* viewpoint, imposing that aggregation can always be decomposed into a sequence of binary operators.

In fact, the key idea of recursiveness, as introduced in [9], is the existence of an alternative representation in terms of an iterative application of binary operators, at each stage taking advantage of the last previous aggregation. Data are therefore being assumed to be aggregated one by one, and each particular arrangement of data will tell us the sequence of items to be aggregated. Hence, recursiveness in [9] assumes that data show a linear structure, although the decision maker can be allowed to re-arrange data, always within a linear structure, as part of a sometimes needed preprocessing data[1].

The following definitions were given in [9].

Definition 1.- Let us denote

$$\pi_n(a_1, a_2, \ldots, a_n) = (a_{\pi_n(1)}, a_{\pi_n(2)}, \ldots, a_{\pi_n(n)})$$

An *ordering rule* π is a *consistent* family of permutations $\{\pi_n\}_{n>1}$ such that for any possible finite collections of numbers, each extra item a_{n+1} is allocated keeping previous items relative positions, i.e.,

$$\pi_{n+1}(a_1, a_2, \ldots, a_n, a_{n+1}) = (a_{\pi_n(1)}, \ldots, a_{\pi_n(j-1)}, a_{\pi_{n+1}(j)}, a_{\pi_n(j)} \ldots, a_{\pi_n(n)})$$

for some $j \in \{1, \ldots, n+1\}$.

Definition 2.- A left-recursive connective rule is a family of connective operators

$$\{\phi_n : [0,1]^n \to [0,1]\}_{n>1}$$

such that there exists a sequence of binary operators

$$\{L_n : [0,1]^2 \to [0,1]\}_{n>1}$$

verifying

$$\phi_2(a_1, a_2) = L_2(a_{\pi(1)}, a_{\pi(2)})$$

and

$$\phi_n(a_1, \ldots, a_n) = L_n(\phi_{n-1}(a_{\pi(1)}, \ldots, a_{\pi(n-1)}), a_{\pi(n)})$$

for all $n > 2$ and some ordering rule π.

Notice that in no way we are imposing a unique binary operator for the whole iterative process. This was in fact the main criticism argued in [17] against the restrictive result obtained by Fung-Fu [13].

Right recursiveness can be analogously defined, and then we can talk about a *recursive rule* when both left and right representations hold for the same ordering rule (we talk about *standard* recursive rules when they are based upon the identity ordering rule, i.e., that rule that keeps the data order). Then it follows (see [6]) that a connective rule $\{\phi_n\}_{n>1}$ is recursive if and only if a set of general associativity equations (in the sense of Mak [16]) hold for each n, once the ordering rule π has been already applied:

$$\phi_n(a_1, \ldots, a_n) = R_n(a_{\pi(1)}, \phi_{n-1}(a_{\pi(2)}, \ldots, a_{\pi(n)})) = L_n(\phi_{n-1}(a_{\pi(1)}, \ldots, a_{\pi(n-1)}), a_{\pi(n)})$$

must hold for all n.

2 Some results on recursiveness.

Some relevant results on recursive rules have been obtained in [7]. In particular, it was proven that assuming certain *regularity* conditions, recursive rules were restricted to some relevant families of aggregation rules (quasi-additive rules among them). Among those *regularity conditions*, the most relevant one was strict monotonicity.

From the following definition given in [6], it was obtained the next result (see [7]).

Definition 3.- A *regular* recursive connective rule is a family of connective operators

$$\{\phi_n : [0,1]^n \to [0,1]\}_{n>1}$$

such that there exists a sequence of binary continuous operators

$$\{L_n : [0,1]^2 \to [0,1]\}_{n>1}$$

and

$$\{R_n : [0,1]^2 \to [0,1]\}_{n>1}$$

verifying the following conditions:

1.1.- If $x' \leq x''$ and $y' \leq y''$, then

$$L_n(x', y') \leq L_n(x'', y'')$$
$$R_n(x', y') \leq R_n(x'', y'')$$

1.2.- If $x' < x''$ and $y' < y''$, then

$$L_n(x', y') < L_n(x'', y'')$$
$$R_n(x', y') < R_n(x'', y'')$$

[1] Re-arrangement of data, in order to be *consistent* implies that, once the relative position of two elements is being fixed, no extra element to be aggregated will change that relative position.

2.1.- If $x' < x''$, then

$$L_n(x', y) < L_n(x'', y), \quad \forall y$$
$$R_n(x', y) < R_n(x'', y), \quad \forall y$$

2.2.- If $y' < y''$, then

$$L_n(x, y') < L_n(x, y''), \quad \forall x$$
$$R_n(x, y') < R_n(x, y''), \quad \forall x$$

3.1.-

$$L_n(x_i, \bar{x}) \neq L_n(x_i', \bar{x}), \forall \bar{x} \in (0, 1)$$

3.2.-

$$R_n(x_i, \bar{x}) \neq R_n(x_i', \bar{x}), \forall \bar{x} \in (0, 1)$$

3.3.-

$$L_n(\bar{x}, x_i) \neq L_n(\bar{x}, x_i'), \forall \bar{x} \in (0, 1)$$

3.4.-

$$R_n(\bar{x}, x_i) \neq R_n(\bar{x}, x_i'), \forall \bar{x} \in (0, 1)$$

4.1.-

$$\begin{aligned} & L_n(0, y') = L_n(0, y'') = 0, \forall y', y'' \\ \Longleftrightarrow\ & L_n(y', 0) = L_n(y'', 0) = 0, \forall y', y'' \end{aligned}$$

4.2.-

$$\begin{aligned} & R_n(0, y') = R_n(0, y'') = 0, \forall y', y'' \\ \Longleftrightarrow\ & R_n(y', 0) = R_n(y'', 0) = 0, \forall y', y'' \end{aligned}$$

4.3.-

$$\begin{aligned} & L_n(1, y') = L_n(1, y'') \quad \forall y', y'' \\ \Longleftrightarrow\ & L_n(y', 1) = L_n(y'', 1) \quad \forall y', y'' \end{aligned}$$

4.4.-

$$\begin{aligned} & R_n(1, y') = R_n(1, y'') \quad \forall y', y'' \\ \Longleftrightarrow\ & R_n(y', 1) = R_n(y'', 1) \quad \forall y', y'' \end{aligned}$$

Theorem 1.- Let

$$\{\phi_n : [0, 1]^n \to [0, 1]\}_{n>1}$$

be a regular standard rule. If ϕ_n is strictly increasing in each coordinate for all $n > 1$, then there exist:

1. $p : [0, 1] \to R^+$, continuous and strictly increasing function,
2. $\{\delta_n : [0, 1] \to R^+\}_{n>1}$, family of continuous and strictly increasing functions, and
3. $\{c_n\}_{n \geq 1}$, sequence of positive real numbers

in such a way that

$$\phi_n(a_1, \ldots, a_n) = \delta_n^{-1} \left(\prod_{j=2}^{n-2} c_j \sum_{k=1}^{n} c_1^{k-1} p(a_k) \right)$$

for all $(a_1, \ldots, a_n) \in [0, 1]^n$ and for all $n \geq 2$, taking $\prod_{j=2}^{\ell} c_j = 1$ whenever $\ell \leq 2$.

Proof: see [7]. ■

But in order to apply the above result we should check the above regularity conditions, which may not be obvious. In what follows, we provide an alternative result to this one, which is based on the following key result due to Aczél [1].

Theorem 2.- Among the functions, continuous, invertible in both variables on a real interval $[\alpha, \beta]$,

$$\begin{aligned} F(x, y) &= l[f(x) + g(y)] \\ H(x, y) &= l[k(x) + h(y)] \\ G(x, y) &= f^{-1}[k(x) + m(y)] \\ K(x, y) &= h^{-1}[m(x) + g(y)] \end{aligned}$$

is the general solution of

$$F(G(x, y), z) = H(x, K(y, z))$$

where f, g, h, k, l, m are arbitrary continuous and strictly monotonic functions.

Proof: see [1], page 312. ■

Theorem 3.- Let

$$\{\phi_n : [0, 1]^n \to [0, 1]\}_{n>1}$$

be a recursive rule. If L_n and R_n are continuous functions being invertible in both variables for all $n > 1$, then there exist:

1. $p : [0,1] \to R^+$, continuous and strictly function,
2. $\{\delta_n : [0,1] \to R^+\}_{n>1}$, family of continuous and strictly functions, and
3. $\{c_n\}_{n\geq 1}$, sequence of positive real numbers

in such a way that

$$\phi_n(a_1,\ldots,a_n) = \delta_n^{-1}\left(\prod_{j=2}^{n-2} c_j \sum_{k=1}^{n} c_1^{k-1} p(a_k)\right)$$

for all $(a_1,\ldots,a_n) \in [0,1]^n$ and for all $n \geq 2$, taking again $\prod_{j=2}^{\ell} c_j = 1$ whenever $\ell \leq 2$.

Proof: Obviously from the definition of

$$\{\phi_n\}_{n>1}$$

the following generalized associativity equation holds:

$$L_n(R_{n-1}(u,v),w) = R_n(u, L_{n-1}(v,w))$$

Therefore, having $(x_1 \ldots, x_n) \in [0,1]^n$, taking $u = x_1$,$v = \phi_{n-2}(x_2,\ldots,x_{n-1})$ and $w = x_n$ assures the above equation. Keeping in mind the above relation, we know from [1] that the solution of the above general associativity equation is basically additive. That is, there exist $\sigma_n, \theta_n, l_n, p_n, q_n, r_n$ continuous and strictly functions over the compact interval [0,1], which verify:

$$\begin{aligned} R_{n-1}(u,v) &= \sigma_n^{-1}(p_n(u) + q_n(v)) \\ L_{n-1}(v,w) &= \theta_n^{-1}(q_n(v) + r_n(w)) \\ R_n(u,b) &= l_n(p_n(u) + \theta_n(b)) \\ L_n(a,w) &= l_n(\sigma_n(a) + r_n(w)) \end{aligned}$$

so l is a strict monotonic function and L_n is invertible in both variables. Then, fixed $z \in [0,1]$, as L_n is invertible then there exists $(x,y) \in [0,1]^2$ such that $L_n(x,y) = z$. Hence,

$$l_n(p_n(x) + \theta_n(y)) = z$$

and l_n is a sobreyective function and invertible. If we now denote $\delta_n = l_n^{-1}$, we have the following equation:

$$\begin{aligned} R_{n-1}(u,v) &= \sigma_n^{-1}(p_n(u) + q_n(v)) \\ L_{n-1}(v,w) &= \theta_n^{-1}(q_n(v) + r_n(w)) \\ R_n(u,b) &= \delta_n^{-1}(p_n(u) + \theta_n(b)) \\ L_n(a,w) &= \delta_n^{-1}(\sigma_n(a) + r_n(w)) \end{aligned}$$

and proof continues as theorem 3.1 in Amo et al [7]. ■

More general theorems can be obtained taking into account alternative results given by Aczél [1].

3 About the underlying structure.

Recursiveness does not impose any restriction on the nature of data. It is just a theoretical assumption. But data set uses to show a particular structure. In this sense, the recursive approach developed in [9] (see also [6, 7]) is in some way assuming that data are organized according to a linear underlying structure, in such a way data can be alternatively aggregated either from the right or from the left (i.e., from *a beginning* or from *the end*). In case we do not take care of the order, recursiveness refers only to an algorithmic property (how to proceed its calculus), because data show an underlying complete graph and any aggregation is possible.

But linear and complete structures are not the only available structures that allow an iterative calculus, as shown in [15]. An interesting case, for example, is that one where data can be represented according to a *circular* structure. The aggregation can still be conceived in a recursive way, but starting anywhere either towards its left or towards its right. Such a circular structure should allow us interesting results in order to characterize aggregation operators.

4 Preference modeling.

In this section we analyze the preference structure from a recursive point of view.

In preference modeling [12], for each pair of alternatives x, y we have four states: x is worse than y ($x > y$), indifference ($x \sim y$), x is better than y ($x < y$) and incomparability ($x \parallel y$).

But as pointed out in [15], the aggregation of the information between these four states can not be done arbitrarily. For example, we can aggregate the degree of x *is worse than* y with the degree of x *is indifferent to* y. Later we can aggregate the above aggregated class with the degree to which x *is better than* y in order to obtain the negation of the incomparability.

But it is not so obvious the meaning of an aggregated class between class $x > y$ and class $x < y$. The decision maker should define in advance an underlying structure, in which the allowed aggregations are explicated (with no restriction in the case of complete graphs).

A standard graph associated to preference structures is the following:

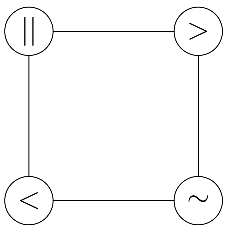

Figure 1: Preference evaluation system

Some of the equations associated to this graph are given by:

- $\Phi_2\left(\mu(x < y), \mu(x \parallel y)\right) = n\left(\Phi_2(\mu(x > y), \mu(x \sim y))\right)$
- $\Phi_2\left(\mu(x < y), \mu(x \sim y)\right) = n\left(\Phi_2(\mu(x > y), \mu(x \parallel y))\right)$
- $\Phi_3\left(\mu(x < y), \mu(x \parallel y), \mu(x > y)\right) = n(\mu(x \sim y))$

for all x, y.

There is a lot of aggregation operators that verify these equations.

Notice that in this case, the four states can be represented as a circular structure.

So we can consider recursive rules as aggregation operators, since *left* and *right* are well defined.

We can also observe that $\Phi_2(a, b) = \Phi_2(b, a)$ for all a, b (symmetry).

If we consider aggregation operators that verify above restrictions assuring quasi-additivity, then we know that there exist p, δ_2, δ_3 and c_1 such that

$$\Phi_2(a, b) = \delta_2^{-1}\left(p(a) + c_1 p(b)\right)$$

and

$$\Phi_3(a, b, c) = \delta_3^{-1}\left(k(p(a) + c_1 p(b) + c_1^2 p(c))\right)$$

Moreover,

$$\Phi_2(a, b) = \Phi_2(b, a)$$
$$\Phi_3(a, b, c) = \Phi_3(c, b, a)$$

Therefore,

$$\delta_2^{-1}\left(p(a) + c_1 p(b)\right) = \delta_2^{-1}\left(p(b) + c_1 p(a)\right)$$

And since δ_2^{-1} is an injective function,

$$p(a) + c_1 p(b) = p(b) + c_1 p(a) \quad \forall a, b \in [0, 1]$$

Then,

$$p(a)[1 - c_1] = p(b)[1 - c_1]$$

So, $c_1 = 1$ and we have that

$$\Phi_2(a, b) = \delta_2^{-1}\left(p(a) + p(b)\right)$$
$$\Phi_3(a, b, c) = \delta_3^{-1}\left(k(p(a) + p(b) + p(c))\right)$$

In other words, when we have a circular structure we can assume the recursivity of the operators, and restrict our model to quasi-additive rules.

The above equations may be helpful in order to obtain membership functions, since the following equations must hold:

$$\delta_2^{-1}\left[p\left(\mu(x < y)\right) + p\left(\mu(x \parallel y)\right)\right] = n\left[p\left(\mu(x > y)\right) + p\left(\mu(x \sim y)\right)\right]$$

$$\delta_2^{-1}\left[p\left(\mu(x < y)\right) + p\left(\mu(x \sim y)\right)\right] = n\left[p\left(\mu(x > y)\right) + p\left(\mu(x \parallel y)\right)\right]$$

$$\delta_3^{-1}\left[p\left(\mu(x > y)\right) + p\left(\mu(x \parallel y)\right) + p\left(\mu(x \sim y)\right)\right] = n\left[p\left(\mu(x < y)\right)\right]$$

$$\delta_3^{-1}\left[p\left(\mu(x < y)\right) + p\left(\mu(x \sim y)\right) + p\left(\mu(x > y)\right)\right] = n\left[p\left(\mu(x \parallel y)\right)\right]$$

$$\delta_3^{-1}\left[p\left(\mu(x \parallel y)\right) + p\left(\mu(x < y)\right) + p\left(\mu(x \sim y)\right)\right] = n\left[p\left(\mu(x > y)\right)\right]$$

$$\delta_3^{-1}\left[p\left(\mu(x < y)\right) + p\left(\mu(x \parallel y)\right) + p\left(\mu(x > y)\right)\right] = n\left[p\left(\mu(x \sim y)\right)\right]$$

These equations can be also taken into account in designing appropriate learning procedures for preference structures.

However, the underlying structure may not be unique.

The above preference structure is indeed behind most standard four-state preference systems (see [10, 11] and [20] but also [8]), showing a circular structure (each vertex of the unit square is only connected with its two adjacent vertices), allowing only certain aggregations, to be obtained by means of an appropriate disjunction operator (we could assume a fixed t-conorm [10], alternatively justified in [20], or even allow disjunction *evolve in time* as in [7, 9]). Aggregation of *non connected* classes should not be considered (see [15]).

Moreover, as pointed out in [4], a conjunction operator will be also needed in order to evaluate the *quality* of the classification system itself, and the whole logical structure should give us hints on how our classification system could be *improved* for future classifications (see [2]).

The need of a learning process for classification may also suggest that perhaps a nice preference structure should include, apart from the above four states $x < y$, $x \sim y$, $x > y$ and $x \parallel y$, a *central* state meaning *undecisiveness* or *ignorance* I, being this extra state connected with each one of other four states: with no information, the whole preference intensity should be associated to such a state, and as we learn more about our preferences, intensities *transfer* between connected states till they are fixed, hopefully assigning no intensity at all to such an *undecisiveness* state.

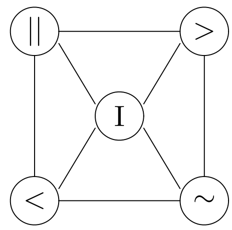

Figure 2: Alternative preference evaluation system

5 Final comments.

The examples above show that we often find out that the family of valuation classes is in many cases *structured* (a graph is being associated to it). The family of valuation classes can vary, the associated underlying structure can be modified, and future changes associated to some *learning process* can be supported by an arbitrary logical structure (not necessarily a standard De Morgan's triple, see [7, 9]). These arguments underlie in [2, 3, 5], where a fuzzy model was considered for the classification of land cover from remotely sensed data: each pixel was classified by means of the whole family of degrees of membership to every class under consideration. As pointed out in [4, 15], a concept should be understood as a structured family of properties, which obviously depend on the context (see also [18, 19]). A recursive aggregation procedure can be then incorporated to our model, in order to allow the aggregated evaluation of adjacent classes (aggregation can not be properly defined for non-adjacent classes).

The model considered in [15] was a particular L-fuzzy set [14], where

$$L = [0,1]^{|\mathcal{C}|}$$

and $\mathcal{C}$ is a structured family of classes (a graph is being defined on it). Once a particular structure has been fixed, each object will be described by means of a vector

$$\mu(x) \in [0,1]^{|\mathcal{C}|}$$

but understanding its meaning needs the associated rules for disjunction, conjunction and negation, to be applied within the particular binary relation defined on $\mathcal{C}$ (see [2, 4], where some measures for relevance, overlapping and redundancy were considered). In this context, the recursive approach proposed in [7, 9] represents an interesting possibility for those connectives, allowing a sequential aggregation of adjacent classes. An important specific case of the model considered in [15] will be Ruspini's partition [21]. Although Ruspini's definition did not assume any particular structure on the family of classes $\mathcal{C}$, we should remind that some kind of underlying structure appears in most cases. For example, the standard 5-valued scale *None, Poor, Average, Very and Complete*,

$$\mathcal{C} = \{ \text{N, P, A, V, C} \}$$

assumes a linear order.

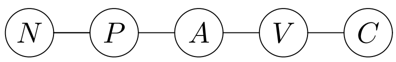

Figure 3: Standard 5-valued evaluation system

Most decision makers have in mind some underlying structure, which of course may not be a complete graph.

Acknowledgments

This research has been supported by the Government of Spain, grant BFM2002-00281. We also thank one of the referees for a carefully reading and suggestions, which indeed improved the paper.

References

[1] J. Aczél (1966): *Lectures on functional equations and their applications.* Academic Press, New York.

[2] A. Amo, D. Gómez, J. Montero and G. Biging (2001): *Relevance and redundancy in fuzzy classification systems.* Mathware and Soft Computing 8, 203–216.

[3] A. Del Amo, J. Montero and G. Biging (1999): Classifying pixels by means of fuzzy relations. *International Journal of General Systems* 29:605–621.

[4] A. Del Amo, J. Montero, G. Biging and V. Cutello (2004): *Fuzzy classification systems.* European Journal of Operational Research, to appear.

[5] A. Del Amo, J. Montero, A. Fernandez, M. Lopez, J. Tordesillas and G. Biging (2002): "Spectral fuzzy classification: an application". *IEEE Trans on Systems, Man and Cybernetics (C)* 32:42–48.

[6] A. Del Amo, J. Montero and E. Molina (2000): *Additive recursive rules*; in J. Fodor *et al.*, eds.: *Preferences and decisions under incomplete knowledge.* Physica-Verlag, Heidelberg.

[7] A. Del Amo, J. Montero, and E. Molina (2001): *Representation of consistent recursive rules.* European Journal of Operational Research 130:29–53.

[8] T. Bilgiç (1998): *Interval-valued preference structures.* European Journal of Operational Research 105:162–183.

[9] V. Cutello and J. Montero (1999): *Recursive connective rules.* International Journal of Intelligent Systems 14:3–20.

[10] J. Fodor and M. Roubens (1994): "Valued preference structures". *European Journal of Operational Research* 79: 277–286.

[11] J. Fodor and M. Roubens (1994): *Fuzzy Preference Modelling and Multicriteria Decision Support* (Kluwer, Dordrecht).

[12] J. Fodor and M. Roubens (1995): *Structure of valued binary relations.* Mathematical Social Sciences 30:71–94.

[13] L.W. Fung and K.S. Fu. (1975): *An axiomatic approach to rational decision making in a fuzzy environment*; in L.A. Zadeh *et al.*, eds.: *Fuzzy sets and their applications to Cognitive and decision processes* (Academic Press, New York); pp. 227–256.

[14] J. Goguen (1967): *L-fuzzy sets*. Journal Mathematical Annals Applications 18: 145–174.

[15] D. Gómez and J. Montero (2003): *Preferences, classification and intuitionistic fuzzy sets*. Proceedings EUSFLAT'03 (University of Applied Sciences, Zittau); pp 187–192.

[16] K.T. Mak (1987): *Coherent continuous systems and the generalized functional equation of associativity*. Mathematics of Operations Research 12:597–625.

[17] J. Montero (1985): *A note on Fung-Fu's theorem*. Fuzzy Sets and Systems 17:259–269.

[18] J. Montero (1986): *Comprehensive fuzziness*. Fuzzy Sets and Systems 20:79–86.

[19] J. Montero (1987): *Extensive fuzziness*. Fuzzy Sets and Systems 21, 201–209.

[20] J. Montero, J. Tejada and V. Cutello (1997): "A general model for deriving preference structures from data". *European Journal of Operational Research* 98: 98–110.

[21] E.H. Ruspini (1969): *A new approach to clustering*. Information and Control 15:22–32.